[in]security

by Gary McGraw

Who Should Do Security?

Bad software lies at the heart of most computer security problems. Yet the people most often tasked with securing an organization aren't software people—they're network operators. Network operators manage the network infrastructure (including the firewalls) that organizations count on for protection. Meanwhile, the software people build custom applications that use protocols such as SOAP, which tunnel traffic through security devices with impunity. (For the most part, I'm talking about custom-built software here. I'll address commercial software in future columns.)

If we're to create systems that can be properly secured, this must change. One way to do that is to include security professionals in every engineering team.

Of course, you'll have to overcome some significant barriers to get operators and builders working together on security. The fact is, the builders themselves aren't pushing for secure solutions; they remain blissfully unaware of their direct impact on security. Trained to think in terms of functions and features, builders think of security as a feature instead of as an emergent system property. Thus, they sprinkle on the magic crypto fairy dust and call it a day. This doesn't work: Protecting communications links with cryptography won't stop input-driven attacks such as the dreaded buffer overflow.

> Security involves more than simply cobbling together disparate security products.

Software security isn't security software. That is, security is a system-wide property that involves more than simply cobbling together disparate security products. It involves subtle behavioral aspects of what many people may think of as nonsecurity-relevant parts of the system. Think of a race condition that happens in a "normal" program function, seemingly unrelated to security. If an attacker can change program behavior and begin to execute a payload, it's game over—regardless of where the hijacking occurs. This is a crucial point that I'll revisit often.

Other barriers to operator and builder cooperation are sociological. The most pervasive problem is the post facto "security review," during which builders are hit with sticks by well-meaning but misguided operations people. Builders are told why the product they busted their humps to complete can't ship. They're told that the security posture can't be changed. And they're generally berated as security imbeciles. This behavior tends to make builders wary of security people.

A better solution is to involve knowledgeable security professionals early in the software life cycle to help anticipate and eradicate problems. These security people must understand software in depth: An inside-out view of software, factoring in software design and code, is always more effective than an outside-in perspective that treats software as a mysterious black box.

Leading departments already have staff devoted to software security. According to an eWeek Webinar poll, 57 percent of security departments have something in place, and another 12 percent are beginning to move security engineering into development. At the very least, close collaboration with the builders in *your* organization is a pressing necessity.

Educational institutions and corporations must make secure coding a priority for all software professionals. This is already beginning to happen: Leading universities such as UC Davis, University of Virginia, Purdue, Johns Hopkins, Stanford, Berkeley, and Princeton are beginning to teach security engineering, and companies such as Qualcomm, Microsoft, and HP have incorporated software security awareness in their training for software types.

Ultimately, operators and builders must learn to get along. Even if they do, software defects with security ramifications—including implementation bugs such as buffer overflows and design flaws such as inconsistent error handling—promise to be with us for years. However, proper security engineering will help.

Gary McGraw is CTO of Cigital, a software quality management consulting provider. He is co-author of *Exploiting Software* (Addison-Wesley, 2004), *Building Secure Software* (Addison-Wesley, 2001), and *Java Security* (Wiley, 1996). Reach him at gem@cigital.com.